A Frequent Term BasedText Clustering Approach Using Novel Similarity Measure

G.Suresh Reddy
Associate Professor
Dept.of IT, VNRVJIET
Hyderabad.
gali.sureshreddy@gmail.com

Dr.T.V.Rajinikanth
Professor
Dept. Of CSE, SNIST
Hyderabad.
rajini_tv@gmail.com

Dr.A.Ananda Rao
Professor and Director IRP
Dept. Of CSE, JNTUA
Anantapur.
akepogu@gmail.com

Abstract— Text clustering is an unsupervised process forming its basis solely on finding the similarity relationship between documents with the output as a set of clusters [14]. In this research, a commonality measure is defined to find commonality between two text files which is used as a similarity measure. The main idea is to apply any existing frequent item finding algorithm such as apriori or fp-tree to the initial set of text files to reduce the dimension of the input text files. A document feature vector is formed for all the documents. Then a vector is formed for all the static text input files. The algorithm outputs a set of clusters from the initial input of text files considered.

Keywords—Commanality measure;frequent item;Clustering; Apriori

I. INTRODUCTION

Text clustering is an unsupervised learning process that is independent of the prior knowledge of data collection or input dataset. There is a strong relation between data mining and machine learning or learnarability.The process of learning may be described as the compression of datasets from the mathematical perspective[5,7].

The relation between complexity of datasets and learnability may be stated mathematically as follows. If we can design an algorithm 'A' which can compress the input dataset, D to a feasible or reduced dataset D' then this means that we have learned some important information. [15-17]. This means that there exists a strong relation between datasets and learnarability.

Dimensionality reduction is also one of the key issues in text clustering and text classification. In the perspective of text mining, dimensionality reduction may be viewed as technique applied to learn and deduce new information from existing input datasets or to achieve learnarability. In this context we can state that learnarability and complexity of datasets are strongly related and hence must be handled very carefully and efficiently.

Let F be a set of all features in a given dataset. Let P be a process of applying dimensionality reduction to dataset D. Now after applying the process P on Dataset D, the feature set F is reduced to F' with reduced number of features or predictive components. In this process of reducing a given dataset, there is also a chance of missing valuable features which may have shown significant contribution in decision making. Such a situation must also be taken care when reducing or eliminating predicate components.

The problem of dimensionality reduction using the concept of finding frequent item or itemsets is gaining significant importance from datamining researchers [18] and this forms basis for the current work. However the method of dimensionality reduction by eliminating stop words, stemming words or using TF-ITF methods has been dealt in the previous works [4].

Clustering is usually carried out using two strategies [3,12]. Incremental Strategy or Complete Strategy also called as Static Strategy. In the Incremental Clustering approach, we may need to re-cluster the text files as and when a new text file is considered for clustering. The main requirement for designing any clustering algorithm is to first design a suitable similarity measure. This may be followed by the design of clustering algorithm or using any of the existing approaches.

In this paper, the process of clustering is carried out by passing the input text files to be clustered to the clustering algorithm. Section II of this paper deals with some of the related works in the literature. In Section III, we discuss the proposed method of clustering the text files and can be extended to any file types. Section-IV concludes the paper.

II. RELATED WORKS

The algorithm designed by Agarwal [8] computes all the frequent items and associations among the frequent items present in a finite static set of transactions. However this is not suitable for dynamically varying database of transactions.

In [1] clustering the given set of text documents from neighbor set is proposed. In [2] the authors propose a method for discovering maximum length frequent item sets. In [6], the classification of text files is performed by using the Gaussian membership function to obtain clusters of text files by computing the word patterns. Each cluster formed is identified by its word pattern computed using Fuzzy Based Gaussian membership function.

In text clustering, the major objective is to get the maximum benefit from the words with in the text file. In this paper we consider clustering text files only.

978-1-4799-2572-8/14/$31.00

However we can apply clustering algorithms to cluster the image files also. This requires converting or transforming the images into numerical data. Since the input to text clustering algorithm is a text corpus, our objective is to transform the text corpus into numerical data, more preferably which can be represented as a numerical vector.

There are several similarity measures defined in the literature such as Euclidean, Cosine, Jaccard, and Manhattan to name a few. For example, the most well known and widely used similarity measure is cosine similarity and Euclidean measures which requires input as numerical vectors. If we are successful in doing so, then we can make use of the existing machine learning algorithms available in the literature or techniques such as text clustering and text classification in data mining [14].

A. Euclidean Measure

The Euclidean measure generally used to find the distance between any points may be used for clustering text documents. Every similarity measure must satisfy certain constraints of similarity measure [10] and Euclidean measure fits effectively. The main disadvantage of Euclidean measure is that it does not take in to consideration the commonality property between the two text files being considered. We may overcome this disadvantage by superimposing commonality feature to Euclidean measure.

B. Cosine Similarity Measure

Another similarity measure which can be used to find similarity measure between any two documents is cosine similarity. The input to cosine similarity is usually vectors. This measure also does not consider the commonality among the documents. This disadvantage is overcome in [10].

C. Tanimoto Coefficient

This is also called as jaccard coefficient. The Jaccard measure considers the commonality among two pair of documents or text files. The function is defined as

$$\text{Jaccard (TextFile1, TextFile2)} = \frac{\text{TextFile1} \cap \text{TextFile2}}{\text{TextFile1 U TextFile2}}$$

The concept of knowledge discovery plays an important role in text mining in the context of machine learning. In machine learning, we train the input data initially to get the initial candidate solution and then we test the new candidate entry with the existing training data set.

The problem of finding similarity between two inputs is one of the important problems in the machine learning. In context of text clustering we can map this problem to find the similarity between two texts or text files. This problem becomes more complex when we try to find similarity in the context of mining social networks [14]. This is because of the unexpected nature of text input which may be typically a function of time sensitivity, texts with very short length, unstructured phrases and redundant information.

III. PROPOSED WORK

A. Problem Definition

To Cluster a given set of text files so as to make text files of similar nature fall into one cluster and those of dissimilar nature into other set of clusters.

A.1 Algorithm Text_Clustering and Topic Identification (N, frequent items, Text files)

// N – number of text files

A.1.1 Preprocessing Phase

Begin

Check if the input file is in .txt or .doc or .docx format. If not, convert it in to proper format

Step1:

For each text file of the form .txt or .doc do

Begin

Step a. Eliminate Stop words

Step b. Eliminate Stemming words

Step c. Apply any frequent item finding algorithm

Step d. Define feature word size equal to size of all Frequent itemsets obtained in step2

End for

End of Preprocessing Phase

B. Processing Phase

Step 2: Form a feature set consisting of m Words consisting of each word in frequent items of each document.

Step 3: Form Binary Matrix with row indicating text file and column each frequent itemset respectively

For each text file in input file set do

Begin

For each frequent item set in feature set do

Begin

If (f_k in feature set W is in text file T_i)

Begin

Define Cell value $M[T_i, F_k] = 1$

// 1 indicates presence of frequent item

Else

Define Cell value $M[T_i, F_k] = 0$

// 0 indicates absence of frequent item

End if

End for

End for

Step 4: Obtain Similarity matrix for each pair of text files applying new similarity Function defined in table 1

Step 5: Count the number of 0's and place the count in the matrix for each pair of text files.

Step 6: Find the cell with maximum value. Group each pair of each such files into a new cluster.

Step 7: Repeat Step6 until no files exist or we reach the stage of first minimum value leaving zero entry.

Step 8: Display all the clusters formed.

Step 9: Identify topics. Give label to each cluster.

End of Processing Phase

For the purpose of finding similarity between two text files, we define the Similarity function S as a function of documents A and B shown below in the truth table.

TABLE I. SIMILARITY FUNCTION OVER FREQUENT ITEMSETS

Frequent item In File1	Frequent item In File 2	Commonality function (file1,file2)
0	0	U
0	1	1
1	0	1
1	1	0

IV. CASE STUDY

A. Finding Frequent items

Intially we start with the text files and reduction the dimension by eliminating stop words, stemming words. After this step we further reduce the dimensions by applying apriori algorithm to this document collection.

TABLE 2. TEXT FILES WITH FREQUENT ITEMS

	List of items after Finding Frequent Items
File1	{sail,earlier,oil,mine,gulf,monday,state,iran}
File2	{sail,earlier,oil,mine,monday,gulf,state,iran}
File3	{state, oil}
File4	{sail,gulf,monday,earlier,state,oil,mine,iran,mail}
File5	{oil}
File6	
File7	{mail}
File8	{mail}
File9	
File10	{state}{mail}

In Table.2, the files denoted by

File1, File2, File4 are crude1.txt, crude3.txt and crude5.txt

File7, File8, File10 are sugar1.txt, sugar2.txt, sugar4.txt

File3 and File5 are crude4.txt, crude6.txt and

File6, File9 are money2.txt and sugar3.txt respectively.

For simplicity, we show the trace of algorithm by considering 10 Sample files from Reuter's dataset -21578. The documents contain the following frequent items as in Table.2

From the above Table II the global frequent items obtained by applying apriori algorithm on all files are given by

Global Features = {Oil, State, Mail, Year, Sail, Earlier, Mine, Gulf, Monday, Iran}.

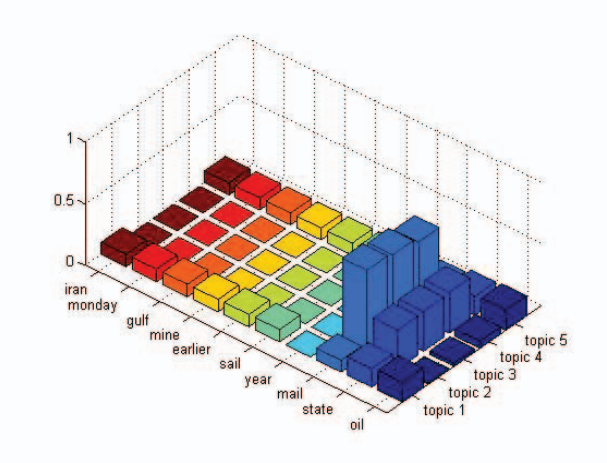


Fig. 1. Distribution of words per topic or cluster for 10 sample files from Reuters-21578 dataset

The distribution of each word for a topic shown in the fig.1 is obtained for the text files considered in table 2 by using matlab version 7.10.0(R2010a).

In Table III we denote global features as w_i where i indicates i^{th} feature.

TABLE III. BINARY REPRESENTATION OF TABLE.2

	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}
F_1	1	1	0	0	1	1	1	1	1	1
F_2	1	1	0	0	1	1	1	1	1	1
F_3	1	1	0	0	0	0	0	0	0	0
F_4	1	1	1	1	1	1	1	1	1	1
F_5	1	0	0	0	0	0	0	0	0	0
F_6	0	0	0	0	0	0	0	0	0	0
F_7	0	0	1	1	0	0	0	0	0	0
F_8	0	0	1	1	0	0	0	0	0	0
F_9	0	0	0	0	0	0	0	0	0	0
F_{10}	0	1	1	1	0	0	0	0	0	0

The Table IV below shows the feature vector computed for each text file pair for one row of the similarity matrix.

TABLE IV. FEATURE VECTOR REPRESENTATION

TEXT_FILE PAIR	FEATURE VECTOR	Count of 0's
[F_1, F_2]	[00UU000000]	8
[F_1, F_3]	[00UU111111]	2
[F_1, F_4]	[00UU000000]	8
[F_1, F_5]	[01UU111111]	1
[F_1, F_6]	[11UU111111]	0
[F_1, F_7]	[1111111111]	0
[F_1, F_8]	[1111111111]	0
[F_1, F_9]	[1111111111]	0
[F_1, F_{10}]	[1111111111]	0

The similarity measure is symmetric, which is the one of the property to be satisfied, hence we can consider only upper diagonal matrix.

The matrix elements are computed by applying the commonality function S for which each document pair forms the input as shown below

TABLE V. SIMILARITY MATRIX OF TEXT FILE PAIRS

	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}
F_1	X	8	2	8	1	0	0	0	0	0
F_2	X	X	2	8	1	0	0	0	0	1
F_3	X	X	X	2	1	0	0	0	0	1
F_4	X	X	X	X	1	0	2	2	0	3
F_5	X	X	X	X	X	0	0	0	0	0
F_6	X	X	X	X	X	X	0	0	0	0
F_7	X	X	X	X	X	X	X	2	0	2
F_8	X	X	X	X	X	X	X	X	0	2
F_9	X	X	X	X	X	X	X	X	X	0
F_{10}	X	X	X	X	X	X	X	X	X	X

For count = 8

The Commonality Value in the cells [F1, F2], [F1, F4], [F2, F4] is 8 .So these files are clustered to form a single cluster [F1, F2, F4] forming Cluster1.

The Commonality Value in the cells [[F7,F8] , [F7,F10],[F8,F10] is 2 .So these files are clustered to form a single cluster [F7,F8,F10] forming Cluster2.

The Commonality Value in the cells [F3, F5] is 1 .So these files are clustered to form a single cluster forming Cluster3.

Finally Cluster4 contains F6 and Cluster5 contains F9.By applying the clustering algorithm over Table.5 we get final set of clusters:

1: [crude1.txt crude3.txt crude5.txt]

2: [sugar1.txt sugar2.txt sugar4.txt]

3: [crude4.txt crude6.txt]

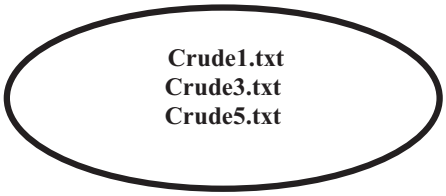
4: [money2.txt]

5: [sugar3.txt]

We can label the clusters for the purpose of identification by using candidate item sets approach followed in [6] or pass it to SVM to classify.

The final set of clusters formed are shown below

CLUSTER1



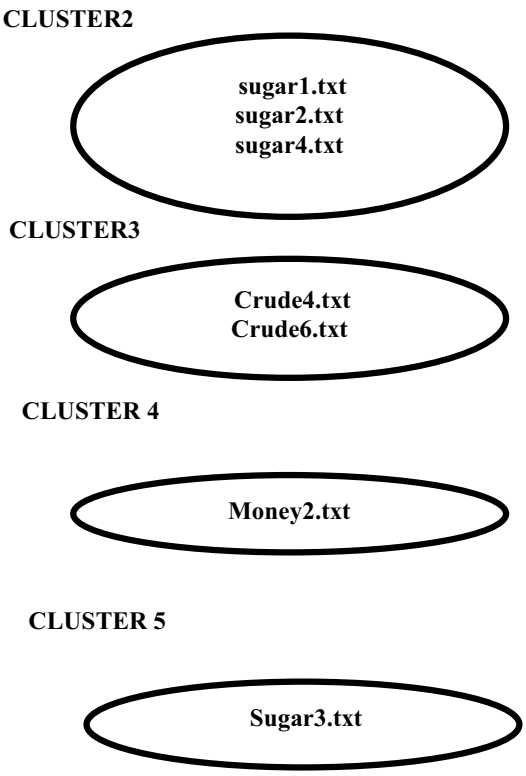


Fig. 2. Clusters Formed for 10 sample files from Reuters-21578 dataset

V. CONCLUSION

The Proposed algorithm has the input as similarity matrix and output a set of clusters as compared to other clustering algorithms that predefine the count of clusters. In this work, frequent items are generated using APRIORI approach by following a similar method. We can replace apriori algorithm by any frequent item finding algorithm. The algorithm for clustering considers the set of frequent items generated from all the documents. This gives the commonality between document pairs. The count of frequent items serves as the distance measure.

REFERENCES

1. Congnan Luo, , Yanjun Li, Soon M. Chung. Text document clustering based on neighbors, Data & Knowledge Engineering (68), 2009, 1271–1288

2. Tianming Hu, Sam Yuan Sung, Hui Xiong, Qian Fu. Discovery of maximum length frequent itemsets, Information Sciences (178), 2008, 69–87

3. Wen Zhanga,, Taketoshi Yoshida, Xijin Tang, Qing Wang. Text clustering using frequent itemsets, Knowledge-Based Systems 23 (2010)379–388

4. Wen Zhanga,Taketoshi Yoshida, Xijin Tang. A comparative study of TF*IDF, LSI and multi-words for text classification. Expert Systems with Applications 38 (2011) 2758–2765

5. Vincent Labatut and Hocine Cherifi. Accuracy Measures for the Comparison of Classifiers, ICIT 2011, The 5th International Conference on Information Technology

6. Jung-Yi Jiang et.al A Fuzzy Self-Constructing Feature Clustering Algorithm for Text Classification, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, VOL. 23, NO. 3, MARCH 2011

7. Melita Hajdinjak, Andrej Bauer. Similarity Measures for Relational Databases, Informatica 33 (2009) 143–149

8. R. Agrawal, T. Imielinski, A. Swami. Mining association rules between sets of items in very large databases, Proceedings of the ACM SIGMOD Conference on Management of data, 1993, pp. 207–216

9. F. Beil, M. Ester, X.W. Xu, Frequent term-based text clustering, in: Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2002, pp. 436–442

10. Yung-shen Lin, Jung-Yi Jiang et.al "A similarity measure for text classification and clustering ", IEEE Transactions on Knowledge and Data Engineering, 2013.

11. Yi Peng, Gang Kou. A Descriptive frame work for the field of data mining and knowledge discovery , International Journal of Information Technology and Decision making, Volume 7, No.4, 2008,Pages 639-682,Impact Factor 3.139

12. Kou,G and Lou,C. Multiple Factor Hierarchical Clustering Algorithm for Large Scale Web Page and Search Engine Clickstream Data, Vol 197,Issue 1,Page 123-134, Annals of Operation Search,2012.

13. Niraj Kumar, Kannan Srinathan. Automatic Key phrase Extraction from Scientific Documents Using N-gram Filtration Technique. Proceedings of the eighth ACM symposium on Document engineering, Pages 199-208, 2008

14. G.Suresh Reddy, Dr.T.V.Rajinikanth, Dr.Ananda Rao Text Clustering Using Frequent Patterns and Jaccard Dissimilarity Function, Proceedings of the second ICACM 2013.

15. Pieter Adriaans and Dolf Zantinge. Data Mining. Eleventh Pearson Education. 2013

16. Daniel Larose. An Introduction to Data Mining. John Wiley Publications

17. Daniel Larose. Data Mining. Models and Methods.2012. John & Wiley Publications

18. K. Ravi Shankar, G.V.R. Kiran, Vikram Pudi. "Evolutionary clustering using frequent itemsets". ACM StreamKDD '10. Proceedings of the First International Workshop on Novel Data Stream Pattern Mining Techniques, Pages 25-30, 2010.